ISBN: 978-0-6480147-3-7

Simplified swarm optimization for feasibility-based rules on constrained engineering design optimization

Wei-Chang Yeh[1] and Chia-Ling Huang[2]

[1]Distinguished Professor, Department of Industrial Engineering and Engineering Management National Tsing Hua University, Hsinchu, Taiwan

yeh@ieee.org

[2]Associate Professor, Department of Logistics and Shipping Management Kainan University, Taoyuan, Taiwan

Corresponding author's E-mail: clhuang@mail.knu.edu.tw

Abstract: *The constrained optimization benchmarks can successfully formulate various practical engineering designs. Therefore, there are various algorithms have been applied to optimize the engineering design by the formulation of the constrained optimization benchmarks since the last decades. The value of the solutions of any objective function that is obtained by any algorithm is ensured only if the solutions of the constraints are falling in feasible solutions. Therefore, it is very important to make sure the solutions are feasible that is the constraints are satisfied before declaring the objective solution obtained by any algorithm. In this work, we use a population-based swarm intelligence algorithm, called simplified swarm optimization (SSO) to optimize the constrained engineering design benchmarks based on the feasible solutions. For the purpose to evaluate the optimization performance, the SSO has performed on three well known constrained engineering design benchmarks including two different types of minimization constrained benchmarks and one engineering constrained and mechanical design benchmark. The computational results have been compared favourably with those obtained using existing algorithms in the literatures. The comparison results demonstrate the proposed SSO well optimize the benchmarks with feasible solutions compared to the other considered algorithms.*

Keywords: simplified swarm optimization (SSO), constrained engineering design, optimization

1. INTRODUCTION

The constrained optimization benchmarks can successfully formulate various practical engineering designs. Therefore, there are various algorithms, such as mine blast algorithm (MBA), harmony search (HS), genetic algorithm (GA), evolutionary programming (EP), improved genetic algorithm (IGA), cultured differential evolution (CA-DE), co-evolutionary particle swarm optimization (CPSO), hybrid particle swarm optimization (HPSO), Nelder-Mead simplex search and particle swarm optimization (NM-PSO), quantum-behaved particle swarm optimization (Q-PSO), and Combining multiobjective optimization with differential evolution (C-DE), have been applied to optimize the engineering design by the formulation of the constrained optimization benchmarks since the last decades [1-15].

As we know, the optimization benchmarks in the real-world practice must be with the constraints. That is all of the optimization engineering design benchmarks are subjected under various constraints. One of the most important techniques is to handle constraints because optimization value obtained by any algorithm seriously losses if the solutions of constraints fall into infeasible regions. Therefore, constraint handling is one of the major concerns when applying the algorithms to solve constrained engineering design optimization benchmarks. First priority, this paper proposes to exam the constraints for feasible solutions or infeasible solutions before comparison the objective solution obtained among each algorithm. Then, the real value of the objective solution obtained by any algorithm can be ensured if having the feasible solutions of constraints.

In this paper, the simplified swarm optimization (SSO) algorithm that belongs to the population-based swarm intelligence is proposed to optimize constrained engineering design benchmarks based on feasible solutions. SSO algorithm is developed and first introduced by Yeh [16] in 2009. SSO has been successfully exploited in considerable studies and applied in various fields. Among these, numerical varieties of constrained engineering design problems have been optimized by SSO, such as optimization of the Disassembly Sequencing Problems [17-19], the series-parallel redundancy allocation problems [20], forecasting wind power [21], and reliability redundancy allocation problems [22].

To demonstrate the performance of SSO, three benchmarks including two different types of minimization constrained benchmarks and one engineering constrained and mechanical design benchmark are considered and demonstrated in this study. The constrained solutions are examined for feasible or infeasible first to ensure the real value of the objective solutions obtained by each algorithm. The experimental solutions are compared with those of previously developed algorithms in literature, the results indicate the proposed SSO performs well for handling feasible solutions of constraints.

2. SIMPLIFIED SWARM OPTIMIZATION (SSO)

SSO is a novel developed soft computing algorithm and first introduced by Yeh [16]. The SSO makes use of a new update mechanism, shown in Eq. (1). The parameter C_w=c_w, C_p= C_w+c_p, and C_g= C_p+c_g, where c_w, c_p, c_g, and c_r are the values of probability; $\rho \in Uniform(0,1)$ and $x \in Uniform(l_j, u_j)$ where l_j and u_j are the lower bound and upper bound in the jth generation for j=1, 2, …, t; solution X_i^t = $(x_{i1}^t, x_{i2}^t, \ldots, x_{i,N_{gen}}^t)$ is a compromise of the current i^{th} solution $X_i^{t-1} = (x_{i1}^{t-1}, x_{i2}^{t-1}, \ldots, x_{iN_{gen}}^{t-1})$, *pBest* $P_i=(p_{i1},p_{i2},\ldots p_{i,\mathrm{Ngen}}) \in \{X_i^1, X_i^2, \ldots, X_i^t\}$ which is a local best such that $F(P_i) \geq F(X_i^j)$, *gBest* $G=(g_1,g_2,\ldots,g_{\mathrm{Ngen}}) \in \{P_1,P_2,\ldots,P_{\mathrm{Nsol}}\}$ which is a global best such that $F(G) \geq F(P_i)$ for i=1,2,…,N_{sol}.

$$x_{ij}^t = \begin{cases} x_{ij}^{t-1} & \text{if } \rho \in [0, C_w) \\ p_{ij} & \text{if } \rho \in [C_w, C_p) \\ g_j & \text{if } \rho \in [C_p, C_g) \\ x & \text{if } \rho \in [C_g, 1) \end{cases} \tag{1}$$

PROCEDURE OF SSO

STEP S0. Generate X_i randomly, let t=1, P_i=X_i, and G=P_j, where $F(P_j) = \underset{i}{Max}\{F(P_i)\}$.

STEP S1. Let i=1.

STEP S2. Update X_i based on Eq. (1) and calculate $F(X_i)$.

STEP S3. If $F(X_i)>F(P_i)$, let $P_i=X_i$; else, go to STEP S5.

STEP S4. If $F(P_i)>F(G)$, then let $G=P_i$.

STEP S5. If $i<N_{sol}$, then let $i=i+1$ and go to STEP S2.

STEP S6. If t=N_{gen}, then G is the final solution and halt; otherwise, let t=t+1 and go to STEP S1.

3. FORMULATION OF THREE BENCHMARKS

The formulations of three benchmarks demonstrated in this study are introduced first in this section.

The First Constrained benchmark

This benchmark is introduced by Brakan and Mccormick [23]. Following, several algorithms are applied to solve this benchmark [1-3, 5].

Two variables are identified and the variable vector is given by

$$\mathbf{x} = (x_1, x_2) \tag{2}$$

The formulation of the first constrained benchmark is displayed as follows:

$$\text{Min} \quad f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \tag{3}$$

$$\text{Subject to} \quad g_1(\mathbf{x}) = x_1 - 2x_2 + 1 = 0, \tag{4}$$

$$g_2(\mathbf{x}) = -(x_1^2/4) - x_2^2 + 1 \geq 0, \tag{5}$$

$$-10 \leq x_i \leq 10, \quad i = 1, 2 \tag{6}$$

The Second Constrained benchmark

The various algorithms are applied to solve the second benchmark [1-2, 4, 6].

In this case, seven variables are identified: x_1, x_2,…, x_7 and the variable vector is given by

$$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \tag{7}$$

The formulation of the second constrained benchmark is displayed as follows:

$$\begin{aligned} \text{Min} \quad f(\mathbf{x}) = {} & (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 \\ & + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 \\ & + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \end{aligned} \tag{8}$$

$$\begin{aligned} \text{Subject to} \quad g_1(\mathbf{x}) = {} & 127 - 2x_1^2 - 3x_2^4 - x_3 \\ & - 3x_4^2 - 5x_5 \geq 0, \end{aligned} \tag{9}$$

$$\begin{aligned} g_2(\mathbf{x}) = {} & 282 - 7x_1 - 3x_2 - 10x_3^2 \\ & - x_4 + x_5 \geq 0, \end{aligned} \tag{10}$$

$$\begin{aligned} g_3(\mathbf{x}) = {} & 196 - 23x_1 - x_2^2 - 6x_6^2 \\ & + 8x_7 \geq 0, \end{aligned} \tag{11}$$

$$\begin{aligned} g_4(\mathbf{x}) = {} & -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 \\ & - 5x_6 + 11x_7 \geq 0, \end{aligned} \tag{12}$$

$$-10 \leq x_i \leq 10, \quad i = 1, 2 \tag{13}$$

The Third Constrained benchmark

Kannan and Kramer [24] propose the third constrained benchmark as shown in Fig. 1, a cylindrical vessel with two hemispherical heads is designed at both ends to minimize the fabrication cost. Several algorithms, such as MBA[1], CPSO[6], HPSO[7], GA1[8], GA2[9], NM-PSO[10], Q-PSO[11] and C-DE[12], are applied to solve this benchmark [1, 6-12].

Four variables are identified: T_s (x_1, thickness of the pressure vessel), T_h (x_2, thickness of the head), R (x_3, inner radius of the vessel), L (x_4, length of the vessel without heads):

$$\mathbf{x} = (x_1, x_2, x_3, x_4) = (T_s, T_h, R, L) \quad (14)$$

The variables T_s and T_h are considered as discrete variables multiples of 0.0625 in, and R and L are treated as continuous variables. The formulation is displayed as follows:

$$\text{Min} \quad f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (15)$$

$$\text{Subject to} \quad g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \le 0, \quad (16)$$

$$g_2(\mathbf{x}) = -x_2 + 0.0954x_3 \le 0, \quad (17)$$

$$g_3(\mathbf{x}) = -\pi x_3^2 x_4 - (4/3)\pi x_3^3 + 1{,}296{,}000 \le 0, \quad (18)$$

$$g_4(\mathbf{x}) = x_4 - 240 \le 0, \quad (19)$$

$$1 \le x_i \le 99, \quad i = 1, 2 \quad (20)$$

$$1000 \le x_i \le 2000, \quad i = 3, 4 \quad (21)$$

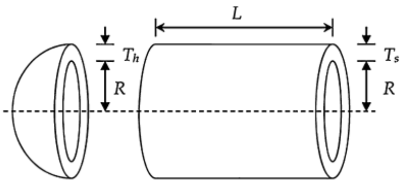

Fig. 1 The pressure vessel design

4. NUMERICAL EXPERIMENTS

The proposed SSO algorithm implemented in all experiments was coded in the C++ programming language and run on an Intel Core i7 3.07 GHz PC with 6 GB memory. The experiments used 1000 generations (N_{gen}=1000), the number of solutions was N_{sol}=100.

Experimental results of the first constrained engineering design benchmark are shown in Table 1.

Table 1. Benchmark 1: Comparison of the solutions

	HS [2]	GA[3]	EP[5]	MBA[1]	Bracken and Mccormick [23]	SSO
x_1	0.8343	0.8080	0.8350	0.82288	0.82288	0.822870567
x_2	0.9121	0.8854	0.9125	0.91144	0.91144	0.911435284
$g_1(x)$	5E−03*	3.7E−02*	1.0E−02*	7.05E−09*	7.05E−09*	**0**
$g_2(x)$	5.4E−03	5.2E−02	−7.0E−02	1.73E−08	1.73E−08	6.7309E-06

f(x)	1.3770	1.4339	1.3772	1.3934649	1.393454	**1.39347741**

In the second row of Table 1, the symbol * means the infeasible solution is obtained. The constraint $g_1(x)$=7.05E−09 which is subjected to equal zero as shown in Eq. (4) falls into infeasible solution leads to the value of objective solution *f(x)*=1.393454 obtained by the algorithm proposed by Bracken and Mccormick [23] seriously loses. The constraint $g_1(x)$ also falls into infeasible solution for the other four algorithms HS [2], GA [3], EP [5], and MBA [1]. Therefore, the optimum solution is *f(x)*= **1.39347741** obtained by the proposed SSO algorithm with the best solution obtained at **x**= (0.822870567, 0.911435284), $g_1(x)$=0, and $g_2(x)$= 6.7309E-06 which are all belonging to the feasible solutions.

Experimental results of the second constrained engineering benchmark are shown in Table 2.

Table 2. Benchmark 2: Comparison of the solutions

	IGA[4]	HS[2]	MBA[1]	CA-DE[13]	SSO
x_1	2.330499	2.323456	2.326585	2.330499	2.17932675
x_2	1.951372	1.951242	1.950973	1.951372	1.879634988
x_3	−0.477541	−0.448467	−0.497446	−0.477541	-0.005798526
x_4	4.365726	4.361919	4.367508	4.365726	4.547563084
x_5	−0.624487	−0.630075	−0.618578	−0.624487	
x_6	1.038131	1.03866	1.043839	1.038131	-0.533158369
x_7	1.594227	1.605348	1.595928	1.594227	1.133152247
$g_1(x)$	4.46E−05	0.208928	1.17E−04	4.46E−05	4.365672767
$g_2(x)$	−252.561723*	−252.878859*	−252.400363*	−252.561723*	0.004486311
$g_3(x)$	−144.878190*	−145.123347*	−144.912069*	−144.878190*	256.0247501
$g_4(x)$	7.63E−06	−0.263414*	1.39E−04	7.63E−06	169.5636351
					32.11470036
f(x)	680.63006	680.6413574	680.6322202	680.6300573	**682.3060712**

In the second row of Table 2, the symbol * means the infeasible solution is obtained. The constraints x_1= −252.561723 and $g_3(x)$= −144.878190 which are subjected to be greater than or equal zero as shown in Eq. (10) and Eq. (11) fall into infeasible solution leads to the value of objective solution *f(x)*= 680.6300573 obtained by the CA-DE algorithm proposed by Becerra and Coello [13] seriously loses. The constraints $g_2(x)$ and $g_3(x)$ also fall into infeasible solution for the other two algorithms IGA [4] and MBA [1]. And the constraints $g_2(x)$, $g_3(x)$ and $g_4(x)$ fall into infeasible solution for the algorithm HS[2]. Therefore, the optimum solution is *f(x)*= **682.3060712** obtained by the proposed SSO algorithm with the best solution obtained at **x**= (2.17932675, 1.879634988, -0.005798526, 4.547563084, -0.533158369, 1.133152247, 4.365672767), $g_1(x)$= 0.004486311, $g_2(x)$= 256.0247501, $g_3(x)$= 169.5636351, and $g_4(x)$= 32.11470036 which are all belonging to the feasible solutions.

Experimental results of the second constrained engineering benchmark are shown in Table 3.

In the second row of Table 3, the symbol * means the infeasible solution is obtained. The constraints x_1= 0.7802 and x_2= 0.3856 which are subjected to be greater than or equal 1 and lesser than or equal 99 as shown in Eq. (20) fall into infeasible solution leads to the value of objective solution $f(x)$= 5889.3216 obtained by the MBA algorithm proposed by Sadollah et al. [8] seriously loses. The constraints x_1 and x_2 also fall into infeasible solution for the other five algorithms GA1[8], GA2[9], CPSO[6], GQ-PSO[11] and C-DE[12]. The constraints x_1, x_2, and $g_3(x)$ fall into infeasible solution for the algorithm HPSO[7]. And the constraints x_1, x_2, and $g_1(x)$ and $g_2(x)$ fall into infeasible solution for the algorithm NM-PSO[10]. Therefore, the optimum solution is $f(x)$= **6008.410926** obtained by the proposed SSO algorithm with the best solution obtained at **x**= (84.99518193, 22.98158617, 172.6715902, 42.63405252), $g_1(x)$= -1.60185E-05, $g_2(x)$= -1.447539968, $g_3(x)$= -0.552476051, and $g_4(x)$= -10.74574313 which are all belonging to the feasible solutions.

Table 3. Benchmark 3: Comparison of the solutions

	GA1[8]	GA2[9]	CPSO[6]	HPSO[7]8	NM-PSO[10]	GQ-PSO[11]	C-DE[12]	MBA[1]	SSO
x_1	0.8125*	0.8125*	0.8125*	0.8125*	0.8036*	0.8125*	0.8125*	0.7802*	84.99518193
x_2	0.4375*	0.4375*	0.4375*	0.4375*	0.3972*	0.4375*	0.4375*	0.3856*	22.98158617
x_3	40.3239	42.0974	42.0913	42.0984	41.6392	42.0984	42.098411	40.4292	172.6715902
x_4	200.0000	176.6540	176.7465	176.6366	182.4120	176.6372	176.637690	198.4964	42.63405252
$g_1(x)$	−3.42E−02	−2.01E−03	−1.37E−06	−8.80E−07	3.65E−05*	−8.79E−07	−6.67E−07	0	-1.60185E-05
$g_2(x)$	−5.28E−02	−3.58E−02	−3.59E−04	−3.58E−02	3.79E−05*	−3.58E−02	−3.58E−02	0	-1.447539968
$g_3(x)$	−304.4020	−24.7593	−118.7687	3.1226*	−1.5914	−0.2179	−3.705123	−86.3645	-0.552476051
$g_4(x)$	−400.0000	−63.3460	−63.2535	−63.3634	−57.5879	−63.3628	−63.362310	−41.5035	-10.74574313
$f(x)$	6288.7445	6059.9463	6061.0777	6059.7143	5930.3137	6059.7208	6059.7340	5889.3216	**6008.410926**

5. CONCLUSION

The SSO algorithm is applied to optimize three constrained engineering design benchmarks. The solutions of the constraints have been examined first as feasible solutions or infeasible solutions before the comparison of the objective solution obtained by each algorithm. Experimental results demonstrate the power of the proposed SSO for dealing with the feasible solutions of constraints. For all of the three constrained benchmarks, the computational results indicate that the proposed SSO obtains feasible solutions for all of the constraints while some infeasible solutions are obtained by the previously developed algorithms in literature [1-13, 23]. The value of the objective solution can be certificated only when the solutions of constraints are feasible solutions. Therefore, the proposed SSO provides optimized objective solutions than other algorithms in terms of examining the solutions of the constraints for feasibility or infeasibility. Moreover, the SSO algorithm can be applied to solve the constrained engineering optimization design especial for which strictly require feasible solutions of the constraints with the accuracy for the objective solution.

Total three famous constrained engineering optimization benchmarks are implemented in this work. In future studies, it will be considered to extend the scale and number of the benchmarks problem.

ACKNOWLEDGMENTS

This research was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C. under grant MOST 104-2221-E-007 -061 -MY3.

REFERENCES

[1] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Applied Soft Computing. 13 (2013) 2592–2612.

[2] K. S. Lee, Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Computer Methods in Applied Mechanics and Engineering. 194 (2005) 3902–3933.

[3] A. Homaifar, C.X. Qi, S.H. Lai, Constrained optimization via genetic algorithms, Simulation. 62 (1994) 242–253.

[4] K. Z. Tang, T. K. Sun, J. Y. Yang, An improved genetic algorithm based on a novel selection strategy for nonlinear programming problems, Computers & Chemical Engineering. 35(4) (2011) 615–621.

[5] D.B. Fogel, A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems, Simulation. 64 (1995) 397–404.

[6] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for engineering optimization problems, Engineering Applications of Artificial Intelligence. 20 (2006) 89–99.

[7] Q. He, L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Applied Mathematics and Computation, 186 (2007) 1407–1422.

[8] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Computers in Industry. 41 (2000) 113–127.

[9] C.A.C. Coello, E. Mezura Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Advanced Engineering Informatics. 16 (2002) 193–203.

[10] E. Zahara, Y.T. Kao, Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Systems with Applications. 36 (2009) 3880–3886.

[11] L.D.S. Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert Systems with Applications. 37 (2010) 1676–1683.

[12] Y. Wang, Z. Cai, Combining multiobjective optimization with differential evolution to solve constrained optimization problems, IEEE Transactions on Evolutionary Computation. 16(1) (2012) 117–134.

[13] R.L. Becerra, C.A.C. Coello, Cultured differential evolution for constrained optimization, Computer Methods in Applied Mechanics and Engineering. 195 (2006) 4303–4322.

[14] Jose Ramirez-Marquez, David Coit, Michael Tortorella, A Generalized Multistate-Based Path Vector Approach for Multistate Two-Terminal Reliability, IIE Transactions. 38(6) (2006) 477-488.

[15] Jose Ramirez-Marquez, David Coit, A Monte-Carlo Simulation Approach for Approximating Multistate Two-Terminal Reliability, Reliability Engineering & System Safety. 87(2) (2005) 253-264.

[16] W.C. Yeh, A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems, Expert Systems with Applications. 36(5) (2009) 9192–9200.

[17] W.C. Yeh, H. Lin, A soft computing algorithm for disassembly sequencing, International Conference on Engineering and Computational Mathematics, 2009.

[18] W.C. Yeh, Optimization of the Disassembly Sequencing Problem on the Basis of Self-Adaptive Simplified Swarm Optimization, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS. 42 (1) (2012) 250–261.

[19] W.C. Yeh, Simplified swarm optimization in disassembly sequencing problems with learning effects, Computers & Operations Research. 39 (2012) 2168–2177.

[20] W.C. Yeh, Orthogonal simplified swarm optimization for the series–parallel redundancy allocation problem with a mix of components, Knowledge-Based Systems. 64 (2014) 1–12.
[21] W.C. Yeh, Y.M. Yeh, P.C. Chang, Y.C. Ke, V. Chung, Forecasting wind power in the Mai Liao Wind Farm based on the multi-layer perceptron artificial neural network model with improved simplified swarm optimization, Electrical Power and Energy Systems. 55 (2014) 741–748.
[22] W.C. Yeh, Y. Z. Jiang, V. Y. Y. Chung, X. He, Solving reliability redundancy allocation problems with orthogonal simplified swarm optimization, International Joint Conference on Neural Networks (IJCNN 2015), Killaney, Ireland.
[23] J. Bracken, G.P. Mccormick, Selected Applications of Nonlinear Programming, John Wiley & Sons, New York, 1968.
[24] B.K. Kannan, S.N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, Journal of Mechanical Design. 116 (1994) 405–411.